

## ЛАБОРАТОРНА РОБОТА № 1

### Принципи програмування. DRY, KISS, SOLID, YAGNI та ін.

**Мета:** навчитися дотримуватися принципів програмування та обґрунтовувати їх.

### Хід роботи:

**Завдання 1:** Виконати завдання з дотриманням відомих Вам принципів програмування. Один на вибір з 3 варіантів.

#### Варіант 2: Зоопарк

1. Створіть систему класів для обліку зоопарку. Ви можете створювати класи для різних видів і підвидів тварин; для вольєрів різних розмірів і типів; корму для тварин; працівників зоопарку.

2. Створіть класи інвентаризації, для виведення на екран інформації про наявних тварин, кількості співробітників тощо.

#### Лістинг програми:

#### Клас Animal:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Animal
    {
        public string Name { get; set; }
        public string Species { get; set; }
        public int Age { get; set; }
        public string Diet { get; set; }

        public Animal(string name, string species, int age, string diet)
        {
            Name = name;
            Species = species;
            Age = age;
            Diet = diet;
        }
    }
}
```

					ДУ «Житомирська політехніка».25.121.30.000 - Лр1			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Якусевич А.І.			Звіт з лабораторної роботи		Літ.	Арк.
Перевір.		Фант М.О.						1
Керівник								12
Н. контр.							ФІКТ Гр. ІПЗ-23-4[2]	
Зав. каф.								

```

public void Eat()
{
    Console.WriteLine($"{Name} їсть {Diet}");
}

public void DisplayInfo()
{
    Console.WriteLine($"{Name} є {Age} років {Species}.");
}
}
}

```

## Клас Employee:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Employee
    {
        public string Name { get; set; }
        public string Position { get; set; }

        public Employee(string name, string position)
        {
            Name = name;
            Position = position;
        }

        public void PerformDuty()
        {
            Console.WriteLine($"{Name} є {Position}, виконує свої обов'язки.");
        }

        public void DisplayInfo()
        {
            Console.WriteLine($"{Name} працює {Position}.");
        }
    }
}

```

## Клас Food:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Food
    {
        public string Name { get; set; }
        public string Type { get; set; }

        public Food(string name, string type)
    }
}

```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        {
            Name = name;
            Type = type;
        }

        public void DisplayInfo()
        {
            Console.WriteLine($"Іжа: {Name}, Тип: {Type}");
        }
    }
}

```

## Клас Inventory:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Inventory
    {
        private List<Animal> animals;
        private List<Employee> employees;
        private List<Food> foods;

        public Inventory()
        {
            animals = new List<Animal>();
            employees = new List<Employee>();
            foods = new List<Food>();
        }

        public void AddAnimal(Animal animal)
        {
            animals.Add(animal);
        }

        public void AddEmployee(Employee employee)
        {
            employees.Add(employee);
        }

        public void AddFood(Food food)
        {
            foods.Add(food);
        }

        public void DisplayInventory()
        {
            Console.WriteLine("Тварини зоопарку:");
            foreach (var animal in animals)
            {
                animal.DisplayInfo();
            }

            Console.WriteLine("\nПрацівники зоопарку:");
            foreach (var employee in employees)
            {

```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

Клас Program:

### Результат виконання програми:

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

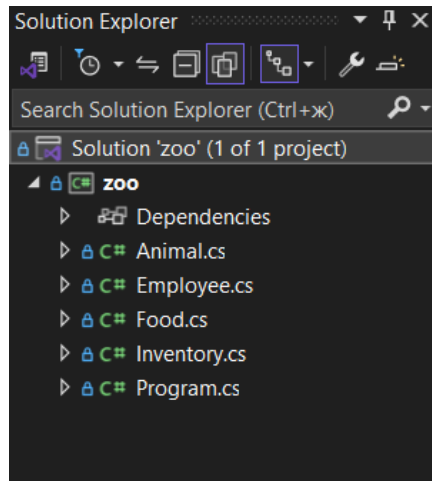


Рис. 1. Структура класів зоопарку

```

Тварини зоопарку:
Сімба є 5 років лев.
Дамбо є 10 років слон.

Працівники зоопарку:
Джон працює Наглядач.
Аліса працює Ветеренар.

Доступна їжа:
Їжа: яловичина, Тип: м'ясо
Їжа: трава, Тип: рослина
Сімба їсть м'ясо
Дамбо їсть трава
Джон є Наглядач, виконує свої обов'язки.
  
```

Рис. 2. Результат виконання програми

**Завдання 2:** Написати код для тестування отриманої функціональності.

1. Покажіть правильність роботи свого коду запустивши його в головному методі програми.
2. Достатньо буде просто вивести певну інформацію, щоб показати, що класи комунікують певним чином між собою.

Лістинг програми:

**Animal**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Animal : IDisplayable
    {
        public string Name { get; set; }
    }
  
```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public string Species { get; set; }
public int Age { get; set; }
public string Diet { get; set; }

public Animal(string name, string species, int age, string diet)
{
    Name = name;
    Species = species;
    Age = age;
    Diet = diet;
}

public Animal(string name, int age)
: this(name, "Невідомий", age, "Невідомий") {}

public void Eat()
{
    Console.WriteLine($"{Name} їсть {Diet}");
}

public void DisplayInfo()
{
    Console.WriteLine($"Тварина: {Name}, Вид: {Species}, Вік: {Age}, Раціон: {Diet}");
}
}

```

## Employee

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Employee : IDisplayable
    {
        public string Name { get; set; }
        public string Position { get; set; }

        public Employee(string name, string position)
        {
            Name = name;
            Position = position;
        }

        public void PerformDuty()
        {
            Console.WriteLine($"{Name} є {Position}, виконує свої обов'язки.");
        }

        public void DisplayInfo()
        {
            Console.WriteLine($"Працівник: {Name}, Посада: {Position}");
        }
    }
}

```

## Food

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Food : IDisplayable
    {
        public string Name { get; set; }
        public string Type { get; set; }

        public Food(string name, string type)
        {
            Name = name;
            Type = type;
        }

        public void DisplayInfo()
        {
            Console.WriteLine($"Корм: {Name}, Тип: {Type}");
        }
    }
}

```

## IDisplayable

```

using System;

namespace zoo
{
    public interface IDisplayable
    {
        void DisplayInfo();
    }
}

```

## Inventory

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace zoo
{
    public class Inventory
    {
        private List<Animal> animals;
        private List<Employee> employees;
        private List<Food> foods;

        public Inventory()
        {
            animals = new List<Animal>();
            employees = new List<Employee>();
            foods = new List<Food>();
        }
    }
}

```

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public void AddAnimal(Animal animal) => animals.Add(animal);
public void AddEmployee(Employee employee) => employees.Add(employee);
public void AddFood(Food food) => foods.Add(food);

public void DisplayInventory()
{
    Console.WriteLine("\n📋 Інвентаризація зоопарку:");

    Console.WriteLine("\n🐾 Тварини:");
    foreach (var animal in animals) animal.DisplayInfo();

    Console.WriteLine("\n👨🔧 Працівники:");
    foreach (var employee in employees) employee.DisplayInfo();

    Console.WriteLine("\n🍖 Корм:");
    foreach (var food in foods) food.DisplayInfo();
}
}

```

## Program

```

using System;
using System.Globalization;
using System.Text;
using zoo;

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = System.Text.Encoding.UTF8;

        Animal lion = new Animal("Сімба", "Лев", 5, "М'ясо");
        Animal elephant = new Animal("Дамбо", "Слон", 8, "Трава");
        Animal giraffe = new Animal("Мелман", "Жираф", 4, "Листя");

        Employee caretaker = new Employee("Олександр", "Доглядач");
        Employee vet = new Employee("Марина", "Ветеринар");

        Food meat = new Food("М'ясо", "Харчування для хижаків");
        Food grass = new Food("Трава", "Їжа для трав'яїдних");

        Inventory zooInventory = new Inventory();
        zooInventory.AddAnimal(lion);
        zooInventory.AddAnimal(elephant);
        zooInventory.AddAnimal(giraffe);
        zooInventory.AddEmployee(caretaker);
        zooInventory.AddEmployee(vet);
        zooInventory.AddFood(meat);
        zooInventory.AddFood(grass);

        zooInventory.DisplayInventory();

        Console.WriteLine("\n✅ Тестування завершено успішно!");
    }
}

```

Результат виконання програми:

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

📦 Інвентаризація зоопарку:

🐾 Тварини:
Тварина: Сімба, Вид: Лев, Вік: 5, Раціон: М'ясо
Тварина: Дамбо, Вид: Слон, Вік: 8, Раціон: Трава
Тварина: Мелман, Вид: Жираф, Вік: 4, Раціон: Листя

👤 Працівники:
Працівник: Олександр, Посада: Доглядач
Працівник: Марина, Посада: Ветеринар

❤️ Корм:
Корм: М'ясо, Тип: Харчування для хижаків
Корм: Трава, Тип: Їжа для травоядних

✅ Тестування завершено успішно!

```

Рис. 3. Результат виконання програми

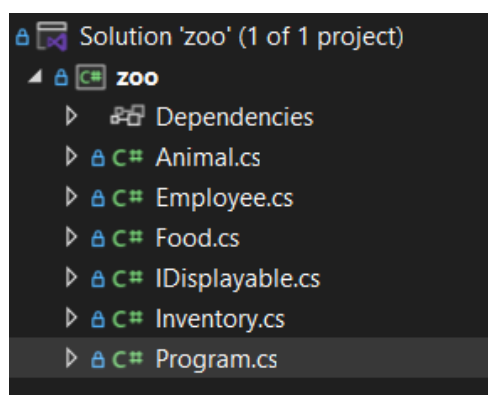


Рис. 4. Структура проєкту

**Завдання 3:** Опишіть особливості дотримання принципів програмування в Вашому коді

1. Додайте файл README.md в кореневу директорію цієї лабораторної роботи. В файлі README.md опишіть дотримання окремо кожного принципу програмування, який Вам відомо, і який можна продемонструвати Вашим кодом.

2. Опис можна залишати українською або (бажано) англійською мовами.

3. Ваш опис повинен містити посилання на відповідні файли і рядки коду.

4. Як залишати посилання на свої рядки коду можна глянути тутечки (для посилання на директорію) або тут (для посилання на окремі рядки).

5. Синтаксис .md файлів документації можна знайти туть або туть.

6. Для отримання максимальної оцінки Ви повинні продемонструвати мінімум 7 принципів. SOLID принципи рахуються окремо. Повний список принципів, які було розглянуто на лекції:

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

- a. DRY,
- b. KISS,
- c. SOLID (5 окремих принципів)
- d. YAGNI
- e. Composition Over Inheritance
- f. Program to Interfaces not Implementations
- g. Fail Fast

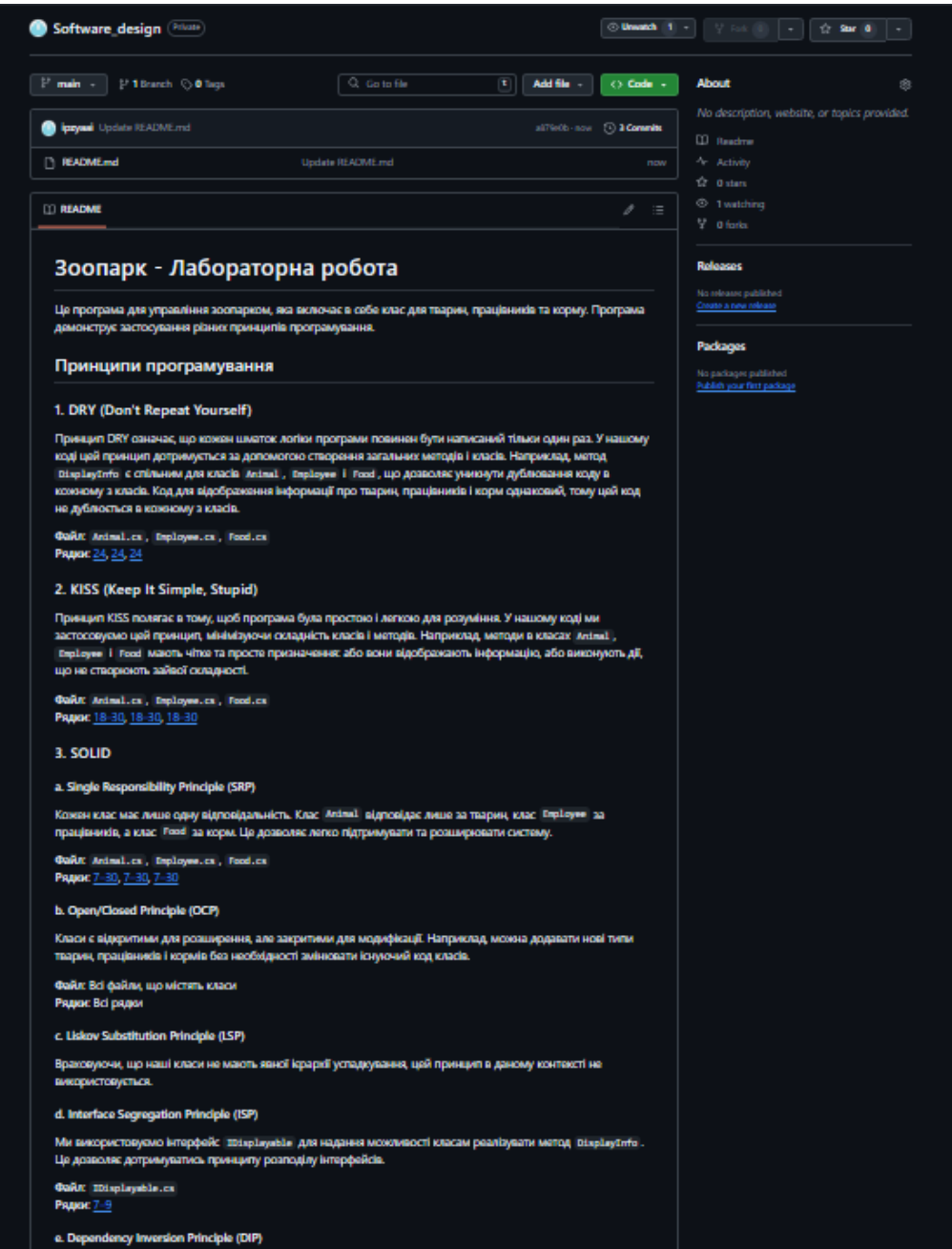


Рис. 5. особливості дотримання принципів програмування пояснення в README.md

#### Завдання 4: UML діаграма

1. Підготувати діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>. Звертайте особливу увагу на стрілки між сутностями. На лекції було розглянуто особливості створення UML діаграм для нашого предмету.

2. Експортувати створену діаграму у вигляді картинки або PDF та запустити експортований файл у кореневу директорію цієї лабораторної.

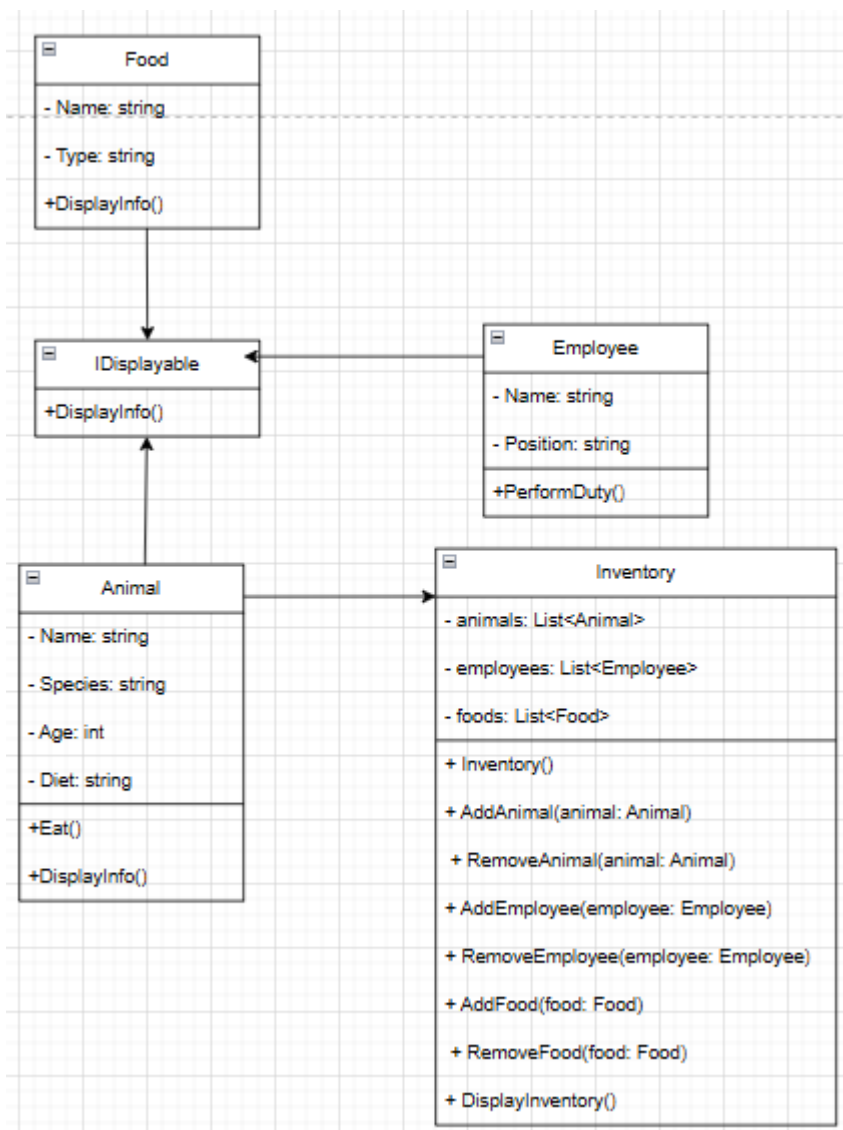


Рис. 6. UML діаграма

**Висновки:** У ході виконання лабораторної роботи було розглянуто та реалізовано основні принципи програмування, такі як DRY, KISS, SOLID, YAGNI, Composition Over Inheritance, Program to Interfaces not Implementations і Fail Fast. Ці принципи допомагають створювати зрозумілий, підтримуваний та масштабований код. Також було розроблено UML-діаграму для наочного

представлення архітектури розробленої системи. Це дало змогу чітко візуалізувати зв'язки між класами та їхню взаємодію. Таким чином, виконана робота дозволила не лише закріпити теоретичні знання, а й отримати практичний досвід у застосуванні принципів програмування, що є критично важливим для розробки якісного та підтримуваного програмного забезпечення.

Посилання на репозиторій: [https://github.com/ipzyaai/Software\\_design.git](https://github.com/ipzyaai/Software_design.git)

		Якусевич А.І.			ДУ «Житомирська політехніка».25.121.30.000 - Лр1	Арк.
		Фант М.О.				12
Змн.	Арк.	№ докум.	Підпис	Дата		